

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

-3-

REMARKS

The Examiner is again thanked for the allowable subject matter.

The Examiner has objected to the Abstract. Such objection is deemed to have been overcome with the amendments made hereinabove.

The Examiner has further rejected Claims 1, 2, 4, 7, 8, 15, and 16 under 35 U.S.C. 103(a) as being unpatentable over Levy (US 5,923,892) of record in the parent application, in view of Marsh (US 5,621,434) of record in the parent application. Applicant respectfully disagrees with such rejection.

It was previously noted that the Examiner relies on the Levy reference, which has an effective date of October 27, 1997. The present application, however, claims the priority of an application filed May 08, 1997. Thus, applicant argued that the use of the Levy reference was improper.

In response to such arguments, the Examiner now argues that "[t]he provisional application 60/045,951 merely consists of an interview with a client about a "hardware invention" and a single drawing. It is not clear how the features claimed in Claims 1-37 are supported by that interview."

In response, applicant provides the following claim charts showing clear support for at least the broadest independent claims rejected above.

Claim 1	Excerpts from Provisional Application 60/045,951
1. An apparatus for accelerating a processor running an object-oriented programming language comprising:	"...and effectively we have this ability to go on, you know, assign tasks to our accelerator function, Java accelerator, and go off and do some other stuff." See page 4, paragraph 4 et al.
a hardware accelerator interfaced with said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and	<p>"...and effectively we have this ability to go on, you know, assign tasks to our accelerator function, Java accelerator, and go off and do some other stuff." See page 4, paragraph 4.</p> <p>"I mean all of the stored code is in ROM, whether it is the OS, the class libraries, our software stubs, everything just resides in ROM." See page 15, paragraph 13 et al.</p> <p>"O.K., well first of all probably the Java applet comes down. We don't do any of the Java by code interpretations, so what happens is it gets sent to the CPU, right, and the CPU has</p>

-4-

	some virtual machine that may be running out of the ROM, right, and then he's storing the Java byte codes at the applets in memory. O.K., so as he is getting it he can start to run this application, so the application is running out of main memory, then it is a shared memory or a [contexture]. And, in the course of this application when it first starts up, it says, O.K., I need to draw my user interface up on the screen now, so he makes a call into the function libraries, which are in ROM, and says O.K., draw me a window" See page 16, paragraph 2 et al.
a software stub that controls interfacing of said hardware accelerator with said processor.	"so our software stub, you know, takes that request, sends out the proper commands to our hardware, our hardware creates a window object and at the same time, goes down to the graphics command generator here and programs, you know, controls the graphic control, you actually draw the window up on the screen." See page 16, paragraph 2 et al.

<b>Claim 4</b>	<b>Excerpts from Provisional Application 60/045,951</b>
1. A method for accelerating a processor running an object-oriented programming language, comprising the steps of:	"...and effectively we have this ability to go on, you know, assign tasks to our accelerator function, Java accelerator, and go off and do some other stuff." See page 4, paragraph 4 et al.
implementing at least one application framework of said object-oriented programming language with a hardware accelerator interfaced with a said processor, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and	"...and effectively we have this ability to go on, you know, assign tasks to our accelerator function, Java accelerator, and go off and do some other stuff." See page 4, paragraph 4 et al.  "I mean all of the stored code is in ROM, whether it is the OS, the class libraries, our software stubs, everything just resides in ROM." See page 15, paragraph 13 et al.  "O.K., well first of all probably the Java applet comes down. We don't do any of the Java by code interpretations, so what happens is it gets sent to the CPU, right, and the CPU has some virtual machine that may be running out of the ROM, right, and then he's storing the Java byte codes at the applets in memory. O.K., so as he is getting it he can start to run this application, so the application is running out of main memory, then it is a shared memory or a [contexture]. And, in the course of this application when it first starts up, it says, O.K., I need to draw my user interface up on the screen now, so he makes a call into the function libraries, which are in ROM, and says O.K., draw me a window" See page 16, paragraph 2 et al.
controlling said interface between said hardware accelerator circuit and said processor with a software stub.	"so our software stub, you know, takes that request, sends out the proper commands to our hardware, our hardware creates a window object and at the same time, goes down to the graphics command generator here and programs, you know, controls the graphic control, you actually draw the window up on the screen." See page 16, paragraph 2 et al.

Thus, applicant respectfully asserts that at least the broadest claims are supported by the provisional application 60/045,951, and the above rejection of at least such claims is improper.

-5-

The Examiner continues by referencing previously submitted affidavits/declarations and states that the "applicant should make the remarks of record in the later application and include a copy of the original affidavit or declaration filed in the parent application." Applicant has satisfied such request, by submitting such declaration, which further overcomes the above rejection. Applicant further notes that such declaration is submitted with a first reply after a final rejection for the purpose of satisfying a requirement made by the Examiner, in which case the declaration is considered timely and should be considered. See MPEP 715.09 (C).

A notice of allowance is respectfully requested.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NVIDP316B).

Respectfully submitted,

Kevin J. Zilka  
Registration No. 41,429

P.O. Box 721120  
San Jose, CA 95172-1120  
408-505-5100

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Poff et al.

Serial No. 08/965,540

Filed: November 6, 1997

For: Hardware Accelerator for an Object-  
Oriented Programming Language

Attorney Docket No. IREA0002

Group Art Unit: 2771

Examiner: U. Le

## DECLARATION UNDER 37 C.F.R. § 1.131

Assistant Commissioner for Patents  
Washington, DC 20231

Sir:

I, John Minami, declare the following:

1. I am a co-inventor of the invention claimed in the above-referenced patent application.
2. PCT application WO 98/21655 has been asserted against the above patent application during the course of examination of the application.
4. The subject matter claimed in the above-referenced patent application is not the same as that set forth in the claims of the above-identified prior art reference WO 98/21655.
6. I enclose herewith a copy of the document "Functional Overview of Java Virtual Machine and Java.NET and Hardware Integration." This document bears the signature of T. Poff as of 13 October 1996. This document is submitted as evidence of Applicant's conception of the claimed invention prior to the effective date of the reference, at least as early as 13 October 1996.
7. Applicant also submits herewith a document entitled iReady Java AWT Engine Specification, version no. 1.00 dated 17 November 1996. This application bears the

Attorney Docket No. IREA0002

-1-

RECEIVED

USSN 08/965,540

JUN 25 1999

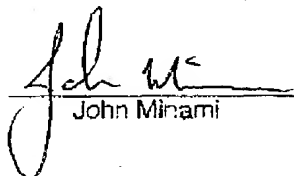
MICHAEL GLENN

signature of T. Poff and the date of 17 November 1996. This document is submitted as evidence of Applicant's due diligence with regard to Applicant's reduction to practice of the invention.

8. Applicant possesses several additional documents which evidence Applicant's continuing diligence in regard to producing the invention to practice which documents will be provided to the Examiner upon request. Applicant notes that the timely completion and filing of a provisional patent application on 8 May 1997 (Application Serial No. 60/045,951) is also further evidence of Applicant's due diligence in reducing the invention to practice.

9. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Date: 6/23/99

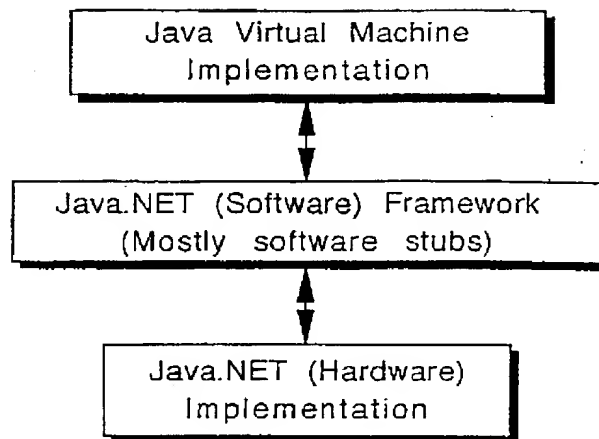
  
John Minami

Attorney Docket No. IREA0002

-2-

USSN 06/965,540

Functional  
Overview of Java  
Virtual Machine  
and Java.NET  
Hardware  
Integration



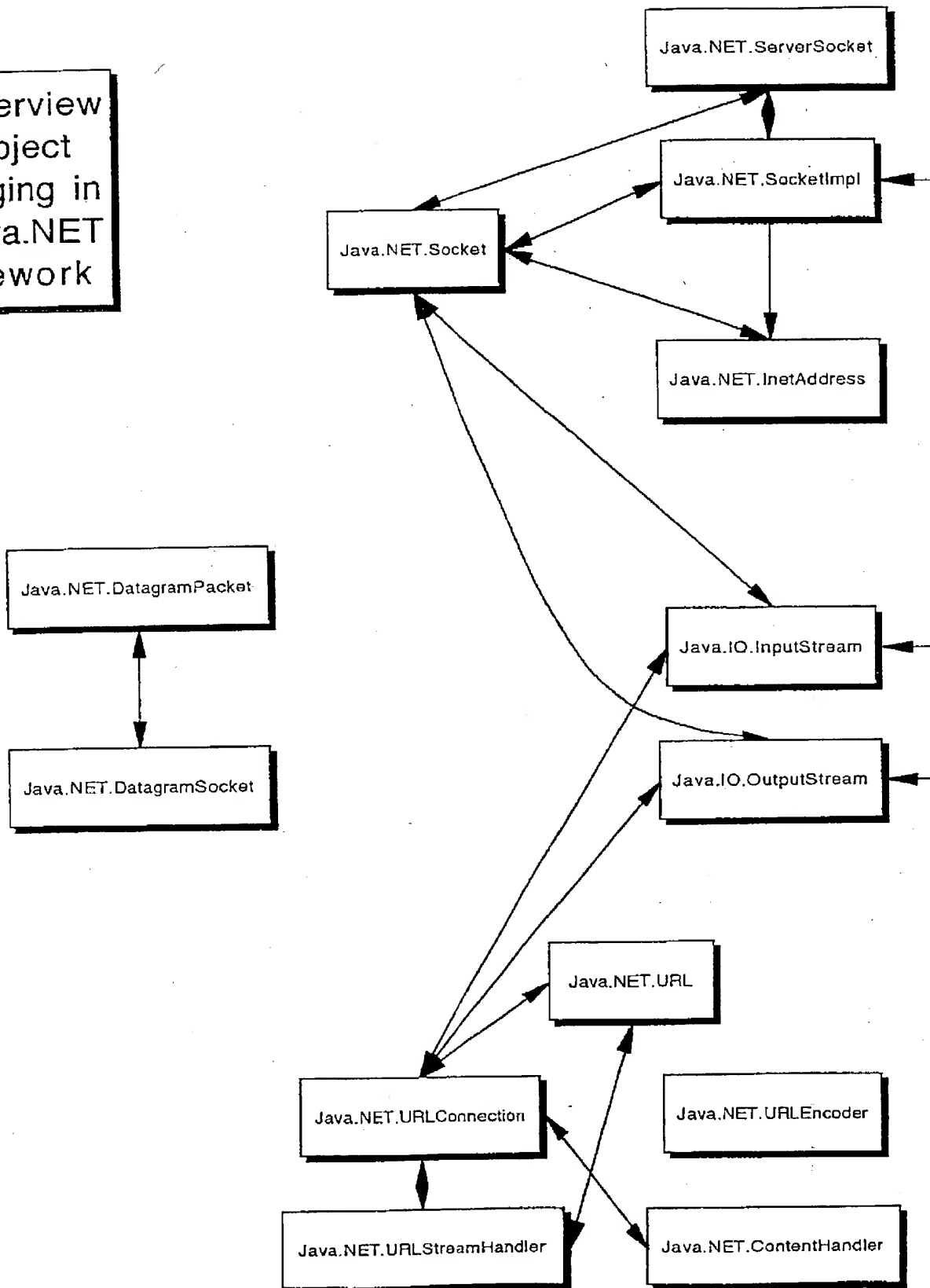
1. Java Virtual Machine Executes java code
2. The JVM runs code within various application frameworks until a request for network connectivity occurs.
3. When a network operation such as a socket request, a read or a write happens and the request is forwarded to the iReady Java.NET Hardware
4. The iReady Java.NET Hardware executes the request. In general, hardware requests will not block program execution in the Java VM Implementation
5. The thread in the VM responsible for performing the request will poll the stack if necessary for data or a return value as necessary.

WITNESSED BY  
*[Signature]*

*Auth Copy  
13 Aug 97*

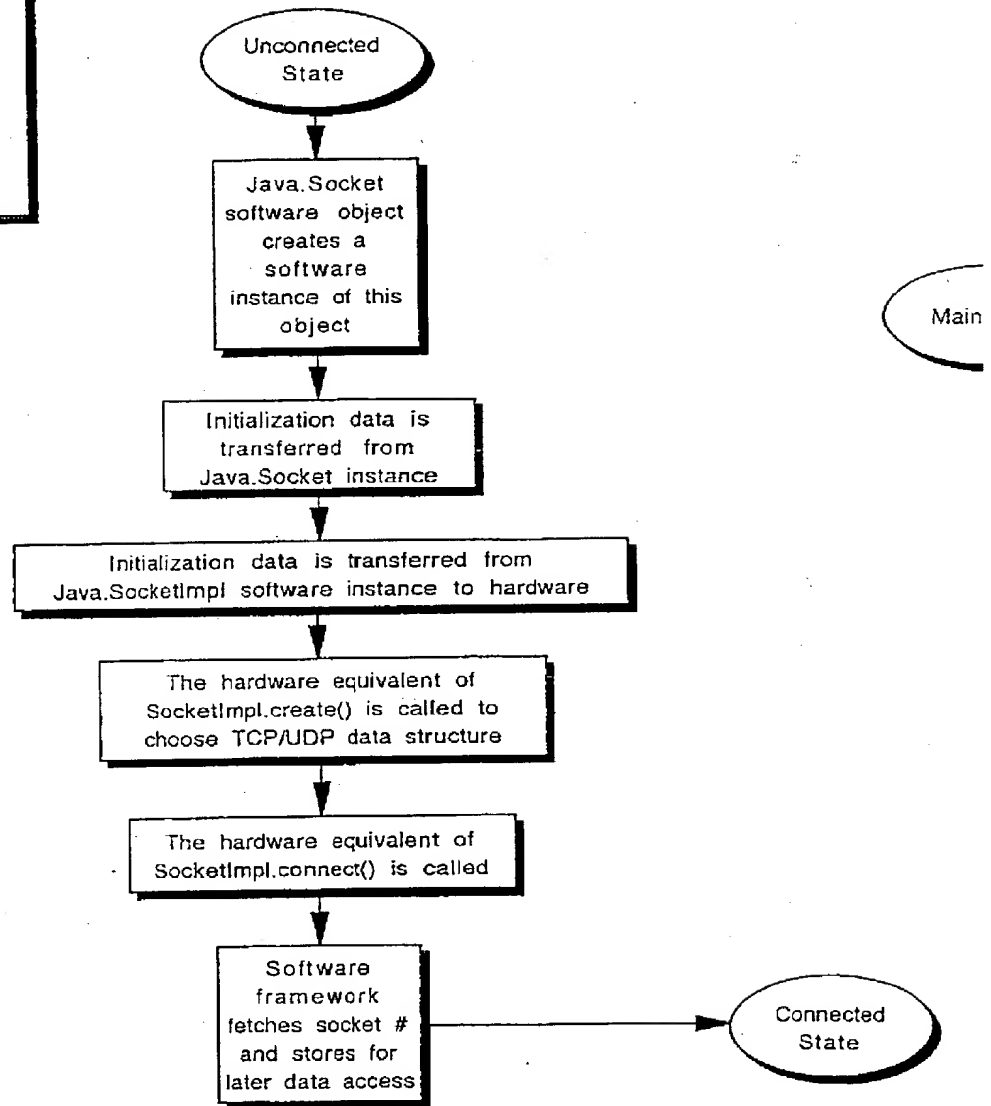
*[Signature]*

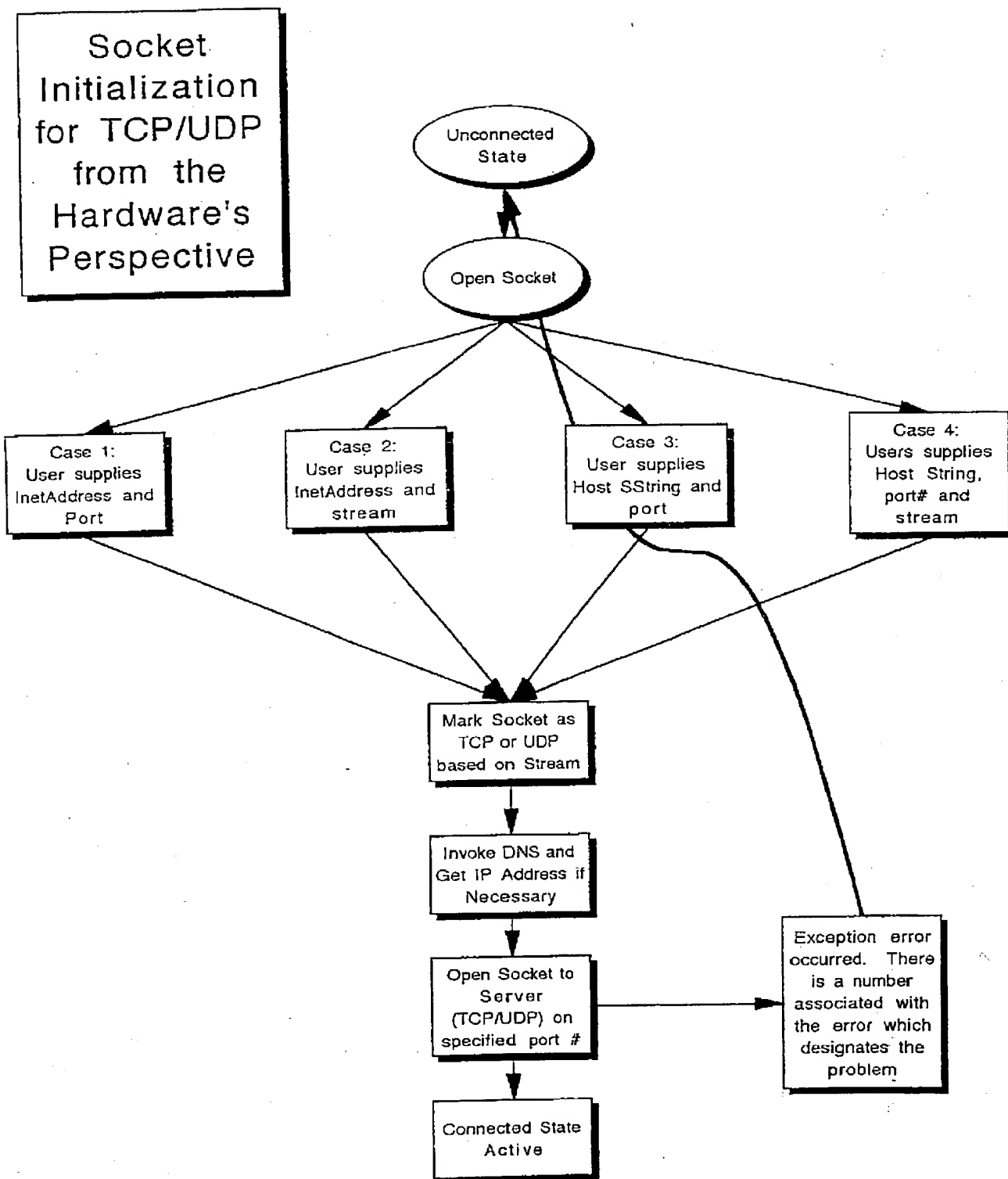
# Overview Object saging in Java.NET mework

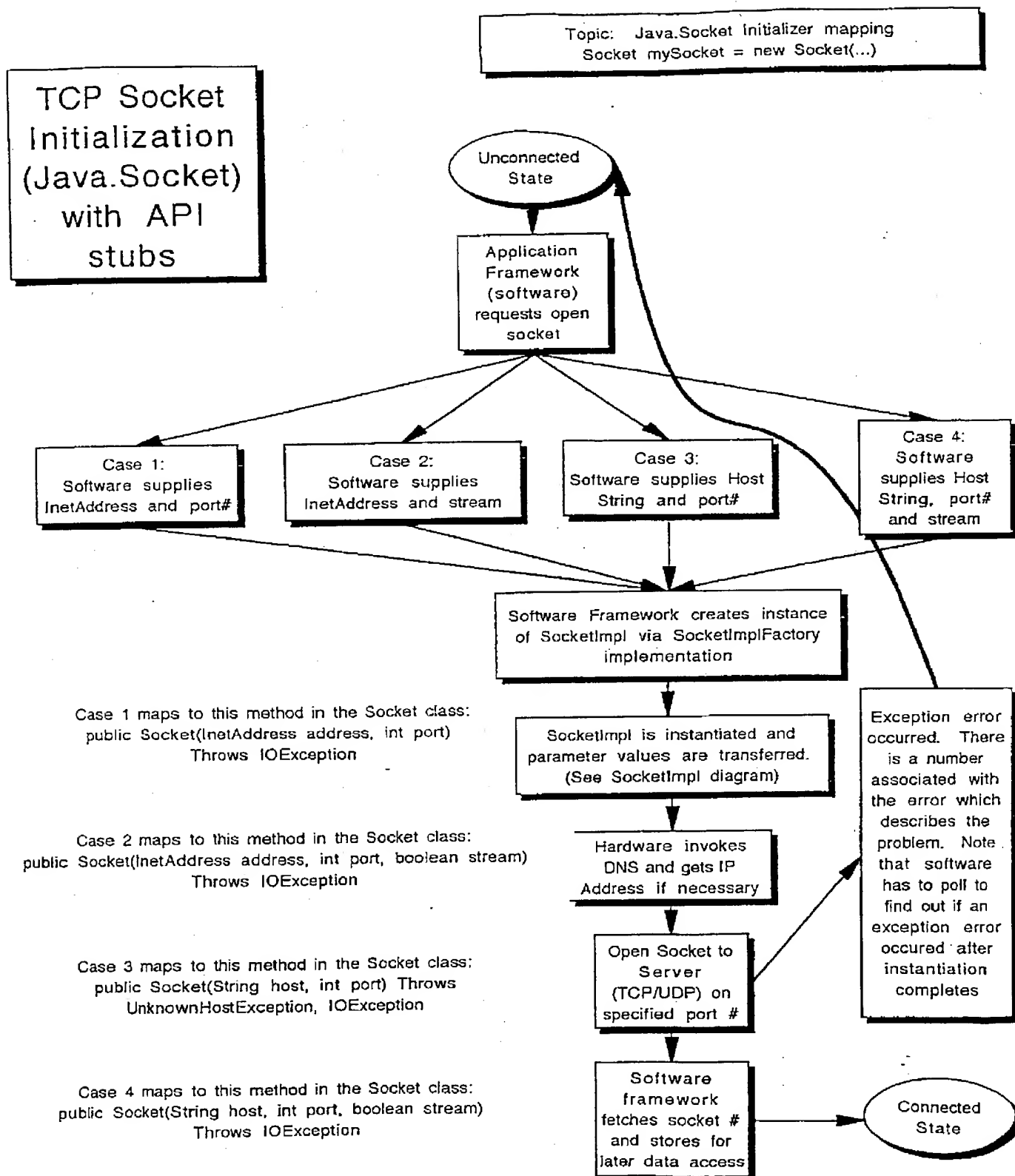




TCP/UDP Socket  
Initialization  
(Java.SocketImpl)  
with API stubs







# TCP Socket Close (Java.Socket) with API stubs

Topic: Java.Socket method mapping  
mySocket.close();

Method Equivalency:  
public void close() throws IOException

Connected  
State

Software  
framework  
wants to close a  
socket

SocketImpl instance (software) associated with  
this Socket instance has its close() method called

Unconnected  
State

Software instance of  
SocketImpl is deallocated

Software instance of  
Socket is deallocated

TCP Socket  
getPort  
(Java.Socket)  
with API  
stubs

Topic: Java.Socket method mapping  
int myPort = mySocket.getPort();

Method Equivalency:  
public int getPort()

Connected  
State

Software framework wants  
to get the local port that it is  
connected on via the  
SocketImpl instance  
associated with this Socket  
instance

Call SocketImpl::getPort() in  
software framework (See  
SocketImpl object definition) and  
return value to caller

Connected  
State